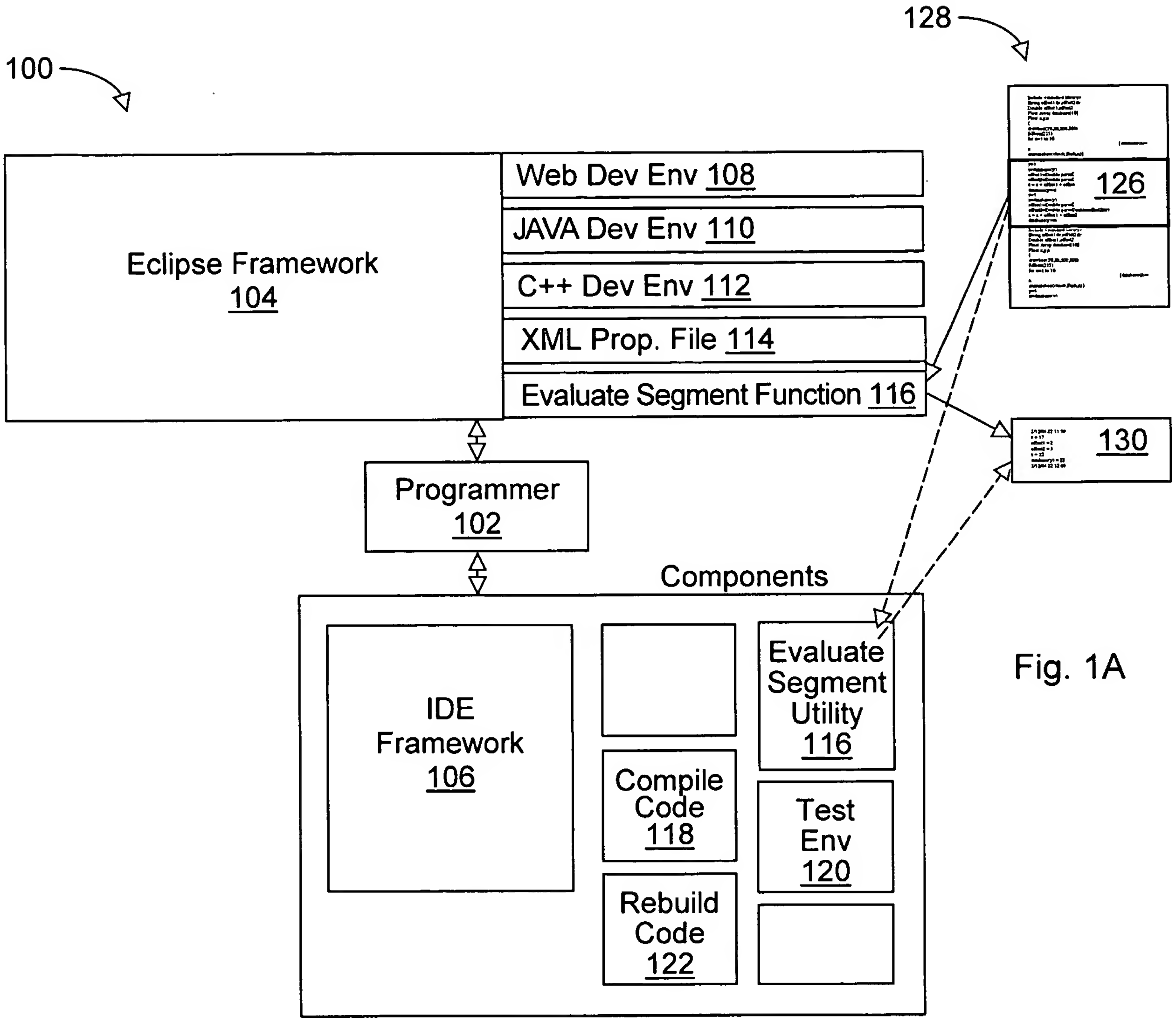

100
128
Eclipse Framework
104
Web Dev Env 108
JAVA Dev Env 110
C++ Dev Env 112
XML Prop. File 114
Evaluate Segment Function 116
126
130
Programmer
102
Components
IDE
Framework
106
Evaluate
Segment
Utility
116
Compile
Code
118
Test
Env
120
Rebuild
Code
122

Fig. 1A

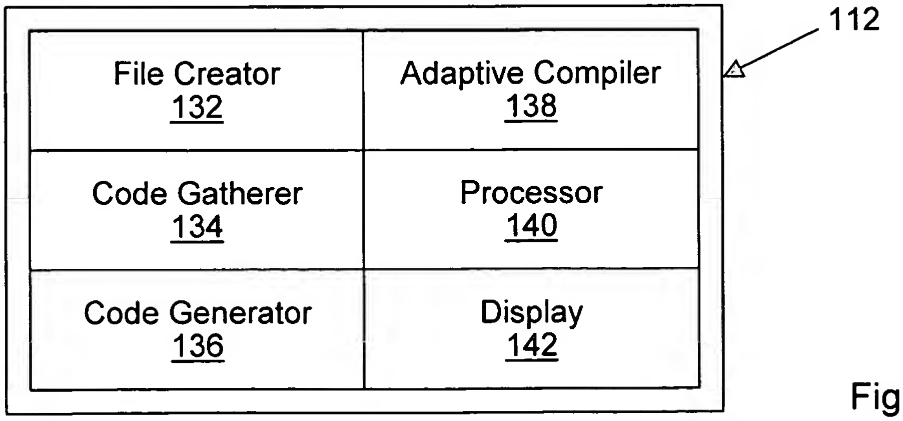

112
File Creator
132
Adaptive Compiler
138
Code Gatherer
134
Processor
140
Code Generator
136
Display
142

Fig. 1B

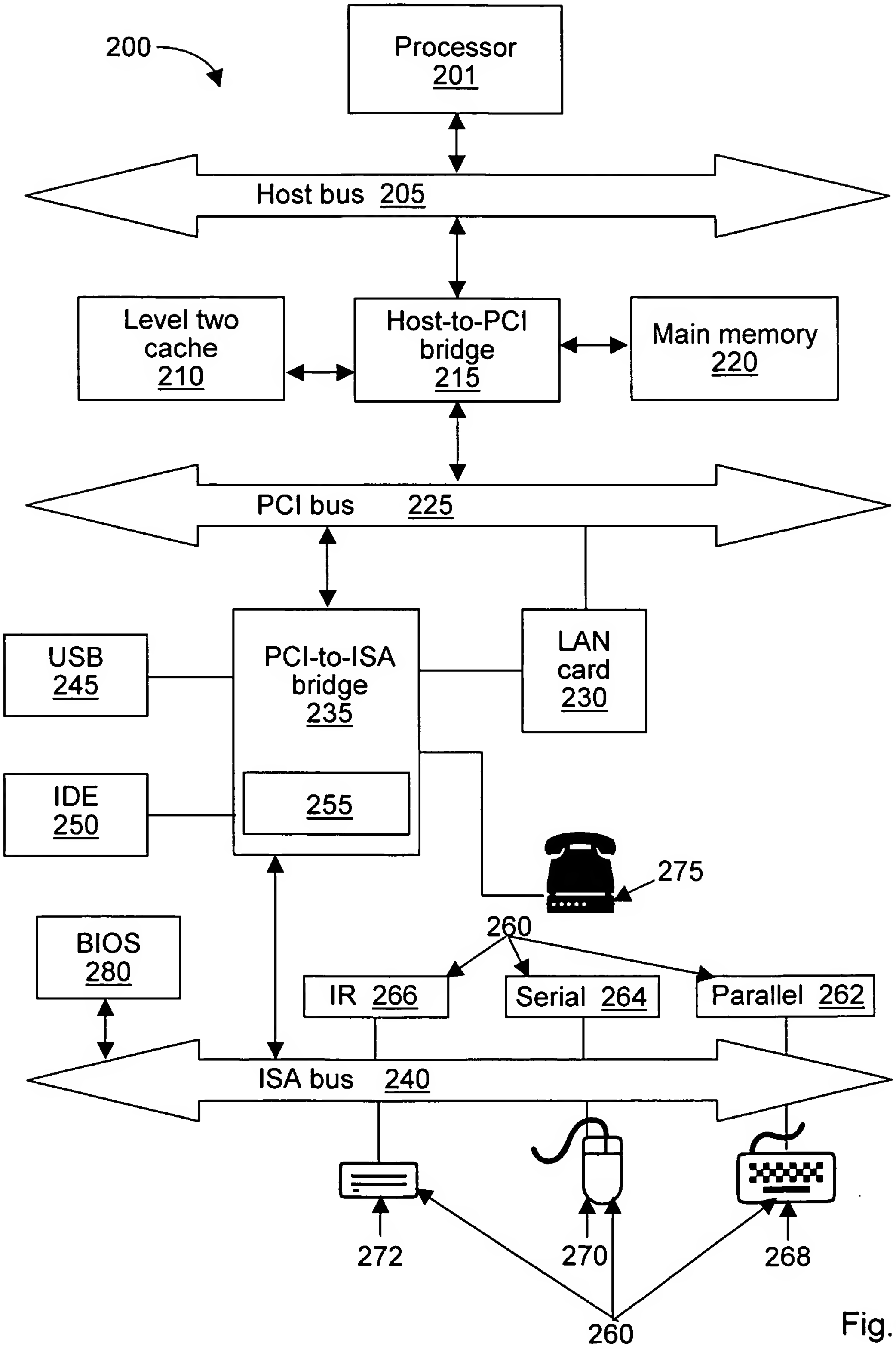
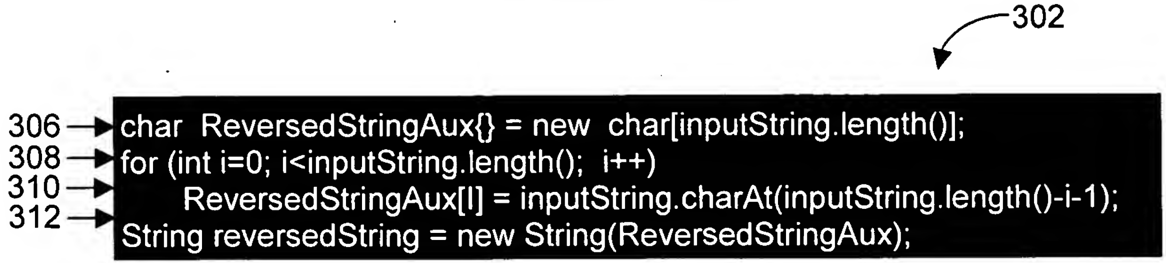
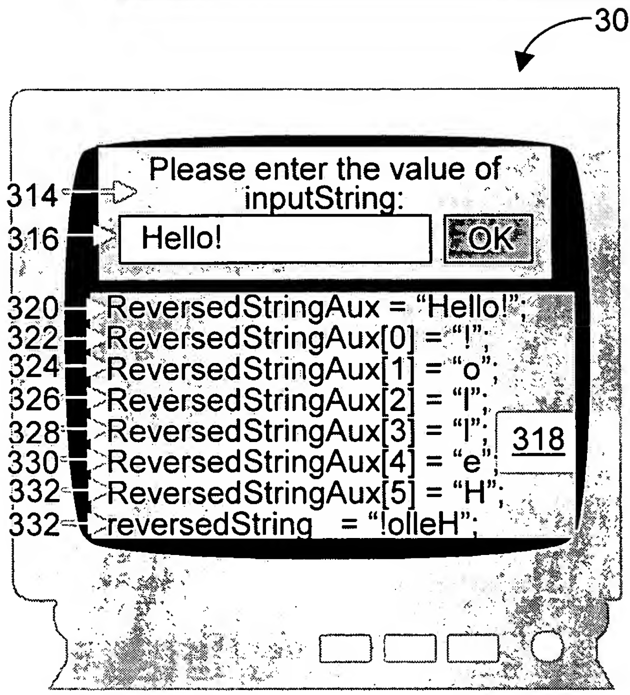

200
Processor 201
Host bus 205
Level two cache 210
Host-to-PCI bridge 215
Main memory 220
PCI bus 225
USB 245
PCI-to-ISA bridge 235
LAN card 230
IDE 250
255
275
BIOS 280
260
IR 266
Serial 264
Parallel 262
ISA bus 240
272
270
268
260

Fig. 2


302
306
308
310
312

```
char  ReversedStringAux{} = new  char[inputString.length()];
for (int i=0; i<inputString.length();  i++)
     ReversedStringAux[I] = inputString.charAt(inputString.length()-i-1);
String reversedString = new String(ReversedStringAux);
```

Fig. 3A

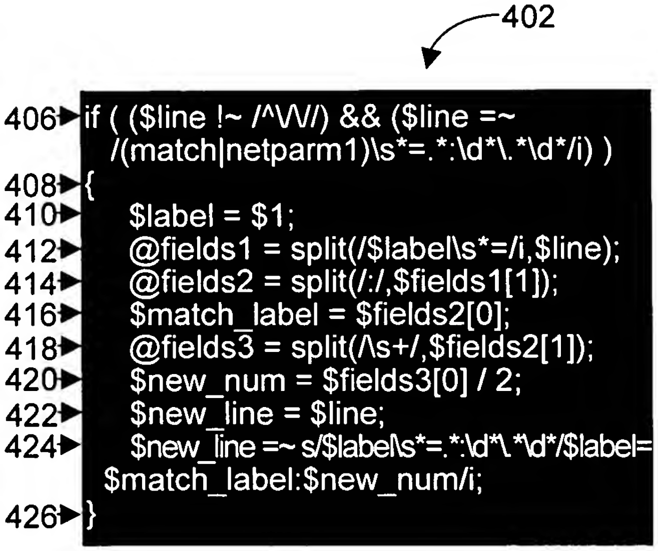


Please enter the value of inputString:

Hello!

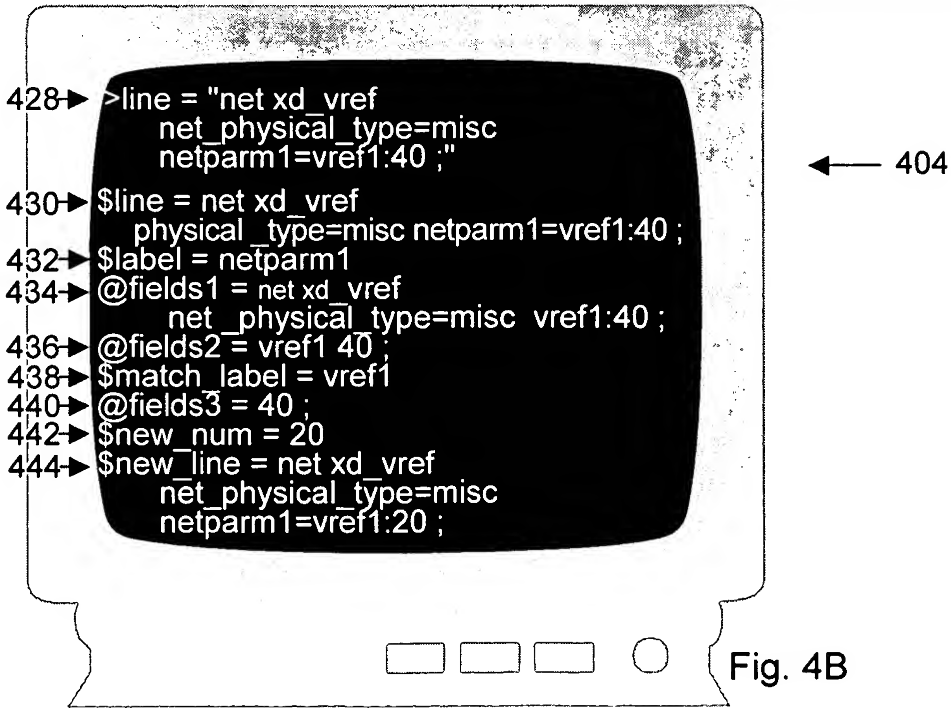
```
ReversedStringAux = “Hello!”;
ReversedStringAux[0] = “!”;
ReversedStringAux[1] = “o”;
ReversedStringAux[2] = “l”;
ReversedStringAux[3] = “l”;
ReversedStringAux[4] = “e”;
ReversedStringAux[5] = “H”;
reversedString   = “!olleH”;
```

Fig. 3B


402
406
408
410
412
414
416
418
420
422
424
426

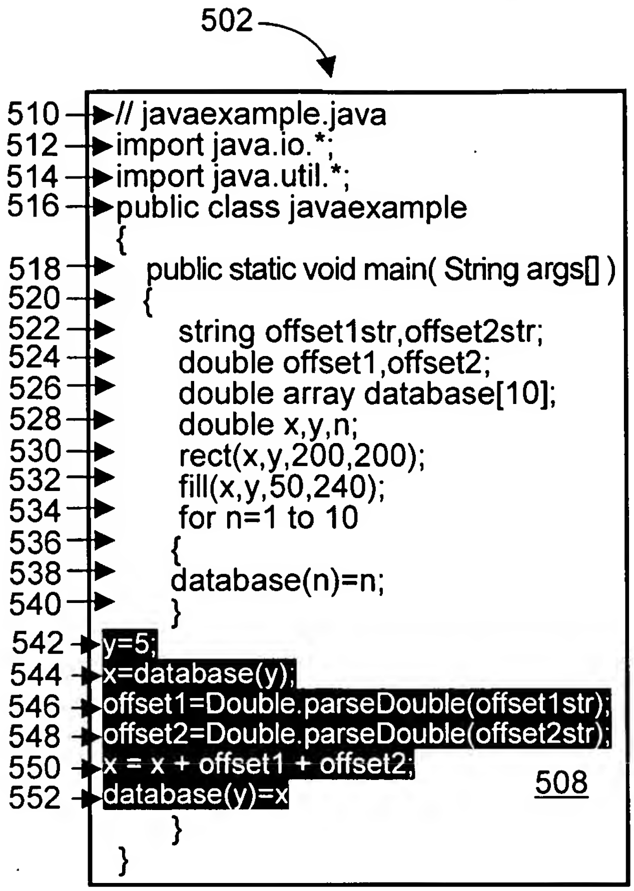
```
if ( ($line !~ /^\W/) && ($line =~
   /(match|netparm1)\s*=.*:\d*\.*\d*/i) )
{
     $label = $1;
     @fields1 = split(/$label\s*=/i,$line);
     @fields2 = split(/:/,$fields1[1]);
     $match_label = $fields2[0];
     @fields3 = split(/\s+/,$fields2[1]);
     $new_num = $fields3[0] / 2;
     $new_line = $line;
     $new_line =~ s/$label\s*=.*:\d*\.*\d*/$label=
  $match_label:$new_num/i;
}
```

Fig. 4A



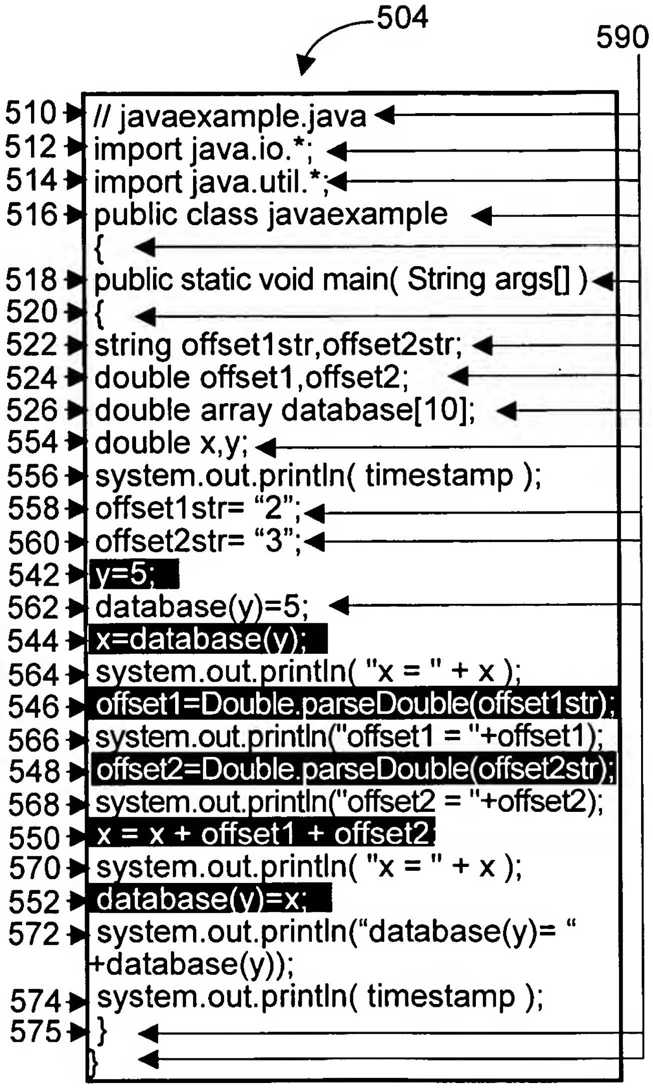
```
>line = "net xd_vref
      net_physical_type=misc
      netparm1=vref1:40 ;"
$line = net xd_vref
     physical _type=misc netparm1=vref1:40 ;
$label = netparm1
@fields1 = net xd_vref
       net _physical_type=misc  vref1:40 ;
@fields2 = vref1 40 ;
$match_label = vref1
@fields3 = 40 ;
$new_num = 20
$new_line = net xd_vref
      net_physical_type=misc
      netparm1=vref1:20 ;
```

Fig. 4B



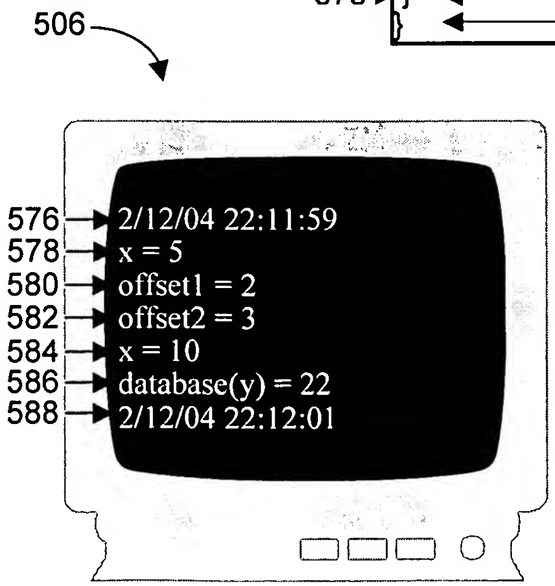
```
510 // javaexample.java
512 import java.io.*;
514 import java.util.*;
516 public class javaexample
    {
518     public static void main( String args[] )
520     {
522         string offset1str,offset2str;
524         double offset1,offset2;
526         double array database[10];
528         double x,y,n;
530         rect(x,y,200,200);
532         fill(x,y,50,240);
534         for n=1 to 10
536        {
538        database(n)=n;
540        }
542 y=5;
544 x=database(y);
546 offset1=Double.parseDouble(offset1str);
548 offset2=Double.parseDouble(offset2str);
550 x = x + offset1 + offset2;
552 database(y)=x
        }
    }
```

Fig. 5A



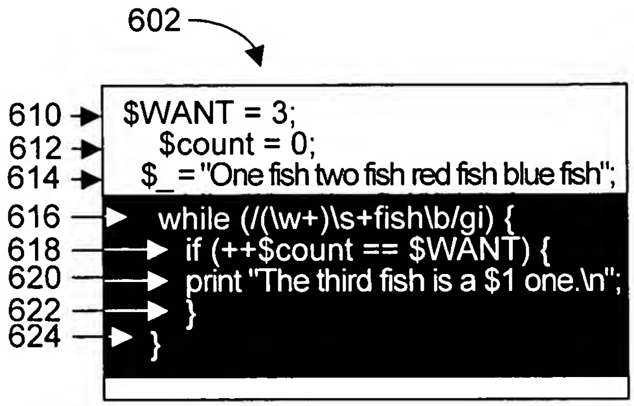
```
510 // javaexample.java
512 import java.io.*;
514 import java.util.*;
516 public class javaexample
    {
518 public static void main( String args[] )
520 {
522 string offset1str,offset2str;
524 double offset1,offset2;
526 double array database[10];
554 double x,y;
556 system.out.println( timestamp );
558 offset1str= "2";
560 offset2str= "3";
542 y=5;
562 database(y)=5;
544 x=database(y);
564 system.out.println( "x = " + x );
546 offset1=Double.parseDouble(offset1str);
566 system.out.println("offset1 = "+offset1);
548 offset2=Double.parseDouble(offset2str);
568 system.out.println("offset2 = "+offset2);
550 x = x + offset1 + offset2
570 system.out.println( "x = " + x );
552 database(y)=x;
572 system.out.println("database(y)= "
    +database(y));
574 system.out.println( timestamp );
575 }
    }
```

Fig. 5B



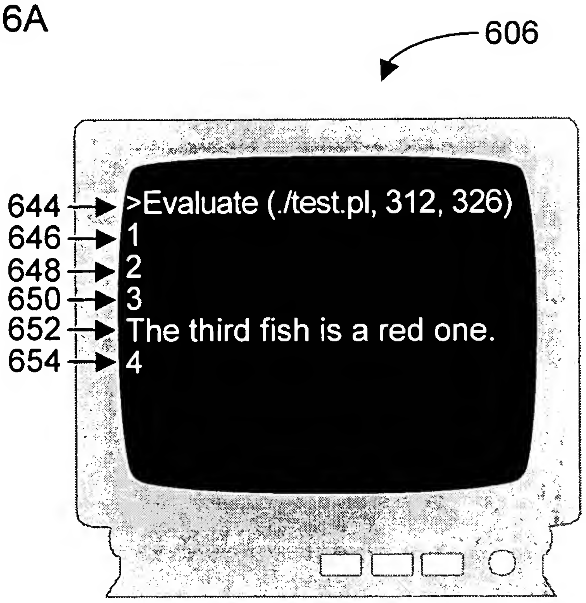
```
576 2/12/04 22:11:59
578 x = 5
580 offset1 = 2
582 offset2 = 3
584 x = 10
586 database(y) = 22
588 2/12/04 22:12:01
```

Fig. 5C



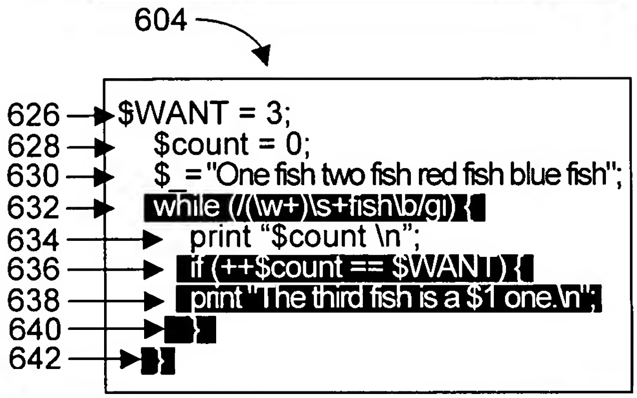
```
$WANT = 3;
  $count = 0;
 $_ = "One fish two fish red fish blue fish";
  while (/(\w+)\s+fish\b/gi) {
    if (++$count == $WANT) {
    print "The third fish is a $1 one.\n";
    }
  }
```

Fig. 6A



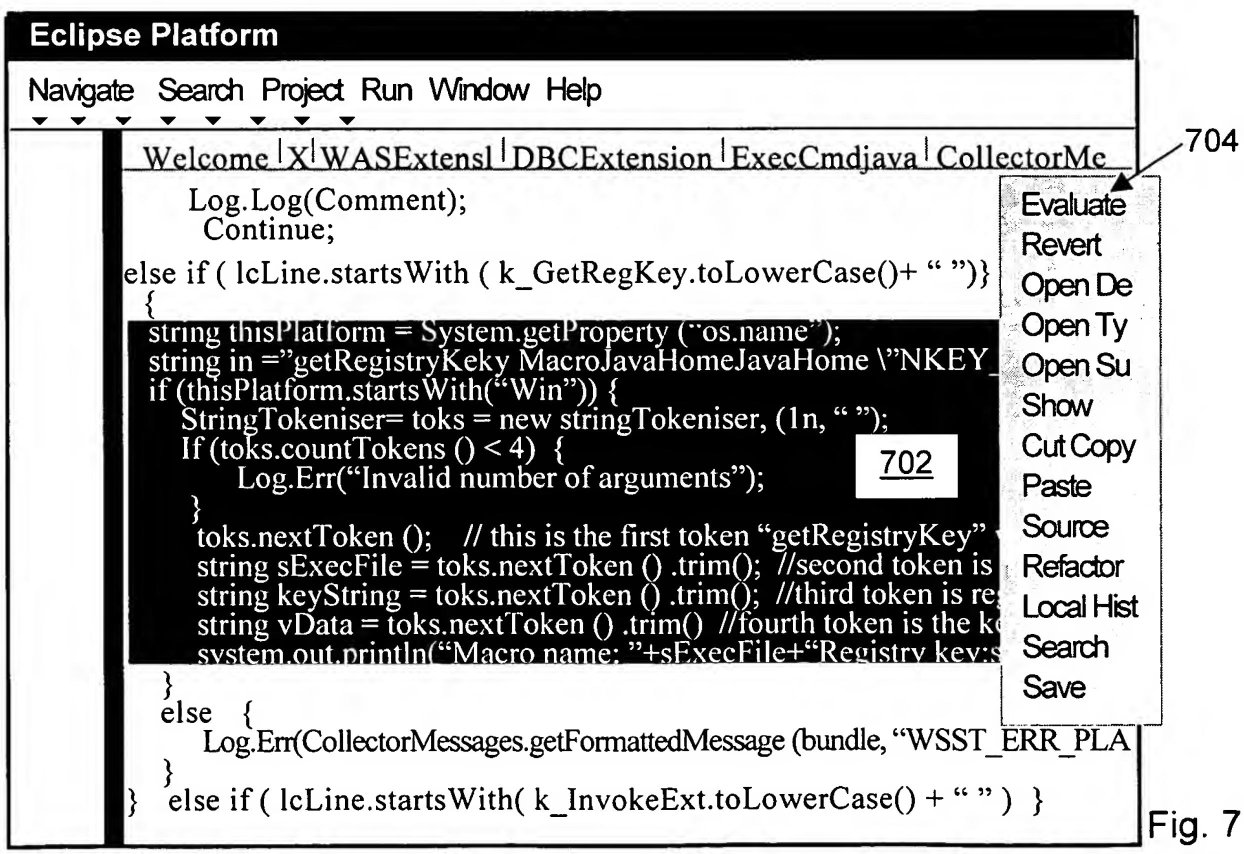
```
>Evaluate (./test.pl, 312, 326)
1
2
3
The third fish is a red one.
4
```

Fig. 6C



```
$WANT = 3;
   $count = 0;
   $_ = "One fish two fish red fish blue fish";
   while (/(\w+)\s+fish\b/gi) {
     print "$count \n";
     if (++$count == $WANT) {
     print "The third fish is a $1 one.\n";
     }
   }
```

Fig. 6B



```
        Log.Log(Comment);
          Continue;
else if ( lcLine.startsWith ( k_GetRegKey.toLowerCase()+ " ")}
  {
  string thisPlatform = System.getProperty ("os.name");
  string in ="getRegistryKeky MacroJavaHomeJavaHome \"NKEY_
  if (thisPlatform.startsWith("Win")) {
      StringTokeniser= toks = new stringTokeniser, (1n, " ");
      If (toks.countTokens () < 4)  {
           Log.Err("Invalid number of arguments");
      }
       toks.nextToken ();    // this is the first token "getRegistryKey"
       string sExecFile = toks.nextToken () .trim();  //second token is
       string keyString = toks.nextToken () .trim();  //third token is re
       string vData = toks.nextToken () .trim()  //fourth token is the k
       system.out.println("Macro name: "+sExecFile+"Registry key:s
   }
   else   {
        Log.Err(CollectorMessages.getFormattedMessage (bundle, "WSST_ERR_PLA
   }
}   else if ( lcLine.startsWith( k_InvokeExt.toLowerCase() + " " )  }
```

Fig. 7

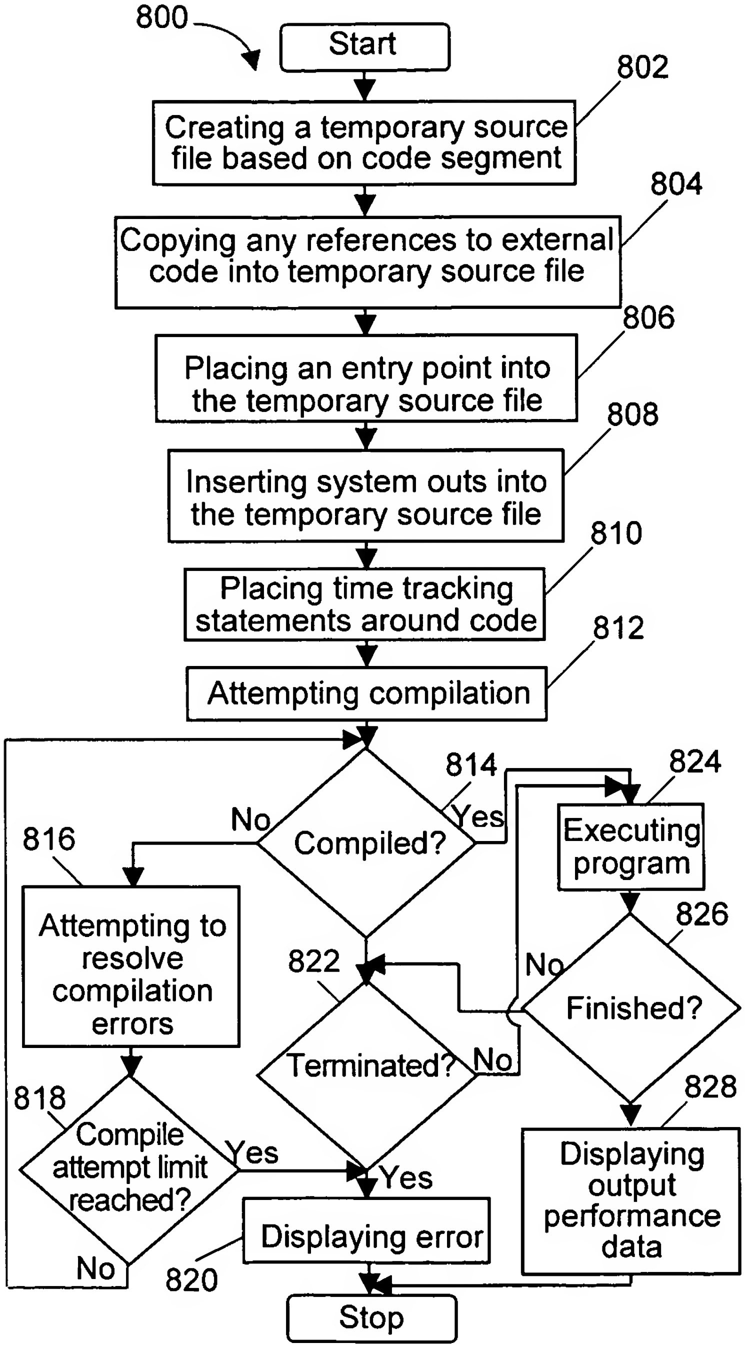

800
Start
802
Creating a temporary source file based on code segment
804
Copying any references to external code into temporary source file
806
Placing an entry point into the temporary source file
808
Inserting system outs into the temporary source file
810
Placing time tracking statements around code
812
Attempting compilation
814
Compiled?
Yes
No
816
Attempting to resolve compilation errors
818
Compile attempt limit reached?
Yes
No
820
Displaying error
822
Terminated?
No
Yes
824
Executing program
826
Finished?
No
828
Displaying output performance data
Stop

Fig. 8

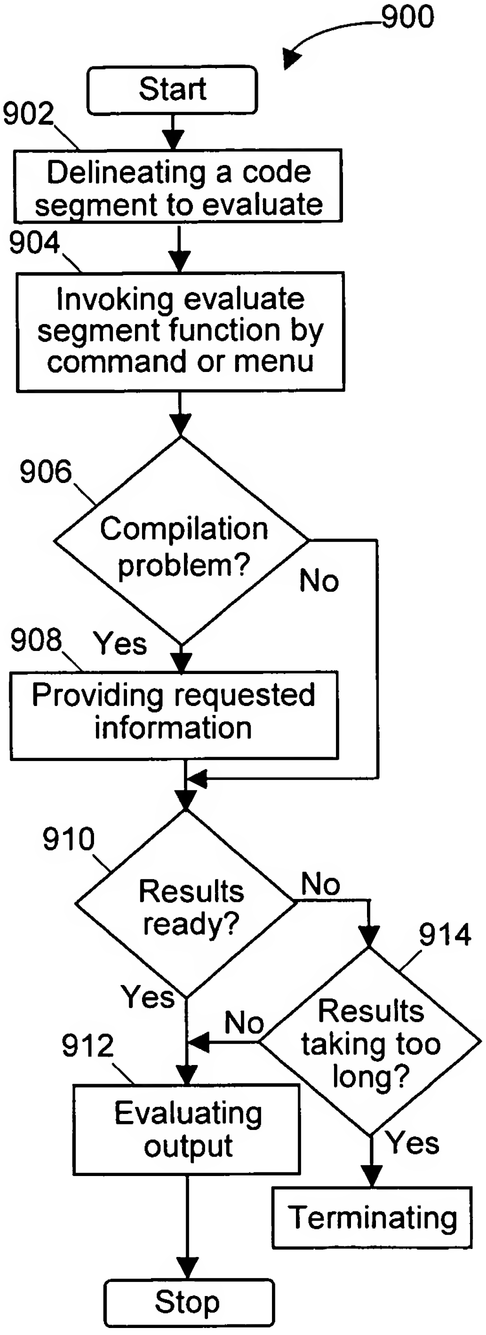

900
Start
902
Delineating a code segment to evaluate
904
Invoking evaluate segment function by command or menu
906
Compilation problem?
No
908
Yes
Providing requested information
910
Results ready?
No
Yes
914
Results taking too long?
No
Yes
912
Evaluating output
Terminating
Stop

Fig. 9